



TRACS: A multi-thread transient current simulator for micro strips and pad detectors

J. Calvo^{a,*}, P. de Castro^a, A. Díez González-Pardo^a, M. Fernández García^b, M. Moll^a,
U. Senica^a, I. Vila^b

^a European Organization for Nuclear Research (CERN), Meyrin, Switzerland

^b Universidad de Cantabria, Instituto de Física de Cantabria (IFCA), Santander, Spain

ARTICLE INFO

Keywords:

Semiconductors
Radiation detectors
RD50
Transient current technique
Ramo current simulator

ABSTRACT

The requisites at the Large Hadron Collider (LHC) at CERN have driven silicon tracking detectors to the fringe of the modern technology. The next upgrade of the LHC to a 10 times increased luminosity of $7.5 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ will require semiconductor detectors with substantially improved radiation hardness. CERN-RD50 collaboration mandate is to provide detector technologies, which are able to operate in such an environment. Within this context, this paper describes the approaches and first results of a C++11 multi-threading software based on the Shockley–Ramo's theorem to simulate non-irradiated and irradiated silicon micro-strips and pad detectors of complex geometries in order to understand signal formation and charge collection efficiencies of arbitrary charge distributions.

1. Introduction

The Large Hadron Collider (LHC) is the largest scientific apparatus ever built. It will remain the most powerful accelerator in the world for at least two decades, and its full exploitation is the highest priority in the European Strategy for Particle Physics [1]. In the second phase of the LHC physics program, the accelerator will provide an additional integrated luminosity of about 2500/fb over 10 years of operation to the general purpose detectors ATLAS [2] and CMS [3]. The experiments will need to address the declining of the present detectors and to improve the ability to isolate and precisely measure the products of the most interesting collisions [4]. In order to face the challenges of unprecedented pp luminosity for detectors, any increase of the radiation hardness and improvement in the understanding of the radiation damage mechanisms will be highly beneficial for the operation of silicon trackers and a possible replacement of pixel layers.

Following these motivations, the CERN-RD50 collaboration [5] (Development of Radiation Hard Semiconductor Devices for Very High Luminosity Colliders) was initiated and approved in June 2002 by the CERN Research Board. A detailed description of the radiation effects is beyond the scope of this article and can be found elsewhere (e.g. Refs. [6,7]). Within the RD50 collaboration, one of the main strategies is material engineering which consists of defect and material characterization as well as computer simulations of defect formation and defect properties, indispensable tools to achieve a profound understanding of the radiation damage process [8]. Along these lines a TRANSient Current Simulator [9], known as TRACS, has been developed as an

extensible tool for providing fast simulations of carrier dynamics inside a semiconductor detector that can have one or several readout electrodes. TRACS has two modes of operation, a graphical user interface (GUI) and a command line interface (CLI). GUI mode is designed for the new user to get a hold on TRACS and for quick checks while the CLI version has more functionalities, it has been designed for more sophisticated and iterative simulations.

Section 2 describes the state of art where TRACS is framed. Section 3 describes the scope and operation modes of the software. Section 4 explains the theoretical background of TRACS, mainly based on Shockley–Ramo's theorem for the calculation of the induced current. In Section 5, numerical methods used to approximate the differential equations are explained and related with the code. Section 6 introduces multi-threading computation in TRACS. Section 7 contains a comparison of TRACS to measured data.

2. State of the art in simulation tools

Exhaustive simulation programs for silicon detectors exist in the market. They can be classified as Technology Computer-Aided Design (TCAD) programs that include the modeling of process steps based on defects and offer a solver for full diffusion and drift equations with additional physics models. Most used TCAD commercial device simulation software are Synopsys Sentaurus [10], Silvaco [11] and Cogenda [12]. They are used to model the structural properties and electrical behavior of semiconductor devices and to predict long term behavior of the detectors.

* Corresponding author.

E-mail address: julio.calvop@gmail.com (J. Calvo).

<https://doi.org/10.1016/j.nima.2018.11.132>

Received 2 April 2018; Received in revised form 18 November 2018; Accepted 28 November 2018

Available online 3 December 2018

0168-9002/© 2018 CERN. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

TCAD tools require full solving of Poisson and continuity equations which implies long simulation time, therefore, using these tools to provide feedback to measurements via, for instance, fits to real data can become unpractical. Few non-commercial and customized fast signal simulators exist, implementing a simple field generation (where traveling charges do not influence the field), imposing a doping profile (instead of deducing it from microscopic defects) or importing a rigorous electric field calculation from TCAD. A first example of these custom simulators is KDetSim [13], a ROOT [14] shared library dedicated to solve Poisson and Laplace equations in 2D and 3D using Monte Carlo simulation of the charge transport inside semiconductor detectors. Another 2D Silicon strip detector simulator is Weightfield2 [15] that allows simulating a silicon strip detector in two dimensions. It determines both drift and weighting fields and calculates the induced current on the readout electrode as a function of the location of a traversing minimum ionizing particle (MIP) or laser beam. A solution for an entire detector using Matlab can be found in [16], for a hybrid detector using SILVACO and Matlab/Simulink in [17] and in [18] for 3D silicon and passive CMOS pixel-detectors.

TRACS solves Laplace and Poisson’s over a mesh representing the detector using specialized and optimized third party finite element methods (FEM) libraries. In comparison with the tools mentioned above, TRACS benefits from parallel computation during the whole execution of iterative simulations, obtaining automatically a better performance of the machine where it is executed, thus, it is particularly advised for CPU intensive simulations combining two or more types of scan, or with fits, where the same calculations are repeated hundreds of times with slight changes of input parameters. Another strong point of TRACS is the possibility to simulate realistic measurement configurations, like an edge scan of a detector for different voltages. A complete explanation of how TRACS works will be given in the next sections.

3. The scope of TRACS

The Transient Current Technique (TCT) is a procedure to determine effective trapping times and field distribution in irradiated silicon detectors. TCT utilizes the drift and diffusion of electron–hole pairs created inside a reverse biased detector by ionizing radiation or laser light. The carriers drift under the influence of an electric field and induce a current in the readout electrodes. The current pulse contains information about the amount of drifting charge and its velocity [19]. TRACS primary goal is to simulate the current induced in microstrip or simple diode (pad) detectors by TCT. Depending on the direction of illumination, TCT can be divided into normal incidence TCT (simply called TCT) or edge-TCT [20] where the detector is probed using a collimated beam from one of the sides, providing information of detector properties as a function of depth. This is illustrated in Fig. 1. Some of the possible uses of TRACS comprise the understanding of laser induced detector responses, simulation of TCT or edge-TCT measurements, calculation of drift velocity, charge collection and electric field profiles in both non-irradiated and irradiated silicon detectors. TRACS also allows to fit the simulation to data and extract detector parameters measurements. Finally, TRACS is also meant to be a flexible function (not only a self-contained program) which can be called as well from an external user program.

3.1. Graphical user interface (GUI)

TRACS GUI, see Fig. 2, is divided into two main parts: the right side allows the user to configure the simulation parameters (size, implant and bulk type, bias, depletion voltage, number and size of strips, end capacitance...) while the left side visualizes the results. The GUI is organized in tabs. Potentials and Fields are solved together and offer both 2D and 1D slices of the distributions. The tab Currents presents a 2D map of the sensor. By double clicking in any point of the map, a pair e–h is generated at the specific position and the current

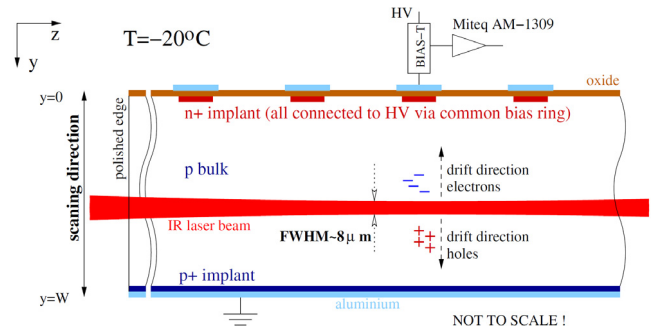


Fig. 1. Schematic view of the edge-TCT technique. Drift of electron/hole pairs injected from the edge of the detector can be studied as a function of the Y coordinate. Source: Image reproduced from [21].

they induce is displayed. This tab also allows to calculate the induced current by arbitrary lines of carriers, specifying start and end points. Finally, the tab Carriers allows to read in an arbitrary distribution of carriers from a file and displays the total current induced. In both tabs, the end capacitance is included as an extra parameter for RC shaping.

3.2. Command line interface (CLI)

A second mode of operation of TRACS consists in using the CLI mode, driven by a user configurable steering file. This file contains the same parameters as the GUI and is extended with further inputs. In CLI mode TRACS behaves as an executable that reads in user parameters and produces ROOT output [14]. The saved data mimics a TCT+ measurement, so the same programs used to analyze measurements can be employed for the simulation.

A C++ interface is finally provided with the software such that a user can create his own executable, using TRACS as a library. This mode extends the flexibility of usage of TRACS to solve more elaborated user problems. This working mode is used, for instance, to extract sensor parameters from fits to measured data.

4. Theoretical background in TRACS

4.1. Principle of operation of Silicon detectors

The most basic model of a silicon detector is given by a simple reverse biased p–n junction [22], where one of the sides of the junction (the implant) is thin and heavily doped compared to the other side, thick and lightly doped (the detector bulk).

Due to the asymmetry of the junction, the depleted width of the thin implant can be discarded, and the depleted width of the detector will only depend on the doping of the bulk. The width of the space charge area $w(V)$ can be calculated integrating Poisson’s equation twice:

$$\omega = \sqrt{\frac{2\epsilon\epsilon_0 V}{q_0 N_{eff}}} \quad (1)$$

In full depletion, the depleted region reaches the entire thickness of the bulk ($w = d$ at the full depletion voltage $V = V_{dep}$):

$$V_{dep} = \frac{q_0 N_{eff} d^2}{2\epsilon\epsilon_0} \quad (2)$$

4.2. Diffusion

In a semiconductor the carriers have a kinetic energy due to their thermal agitation, the average value being $3/2kT$, where k is the

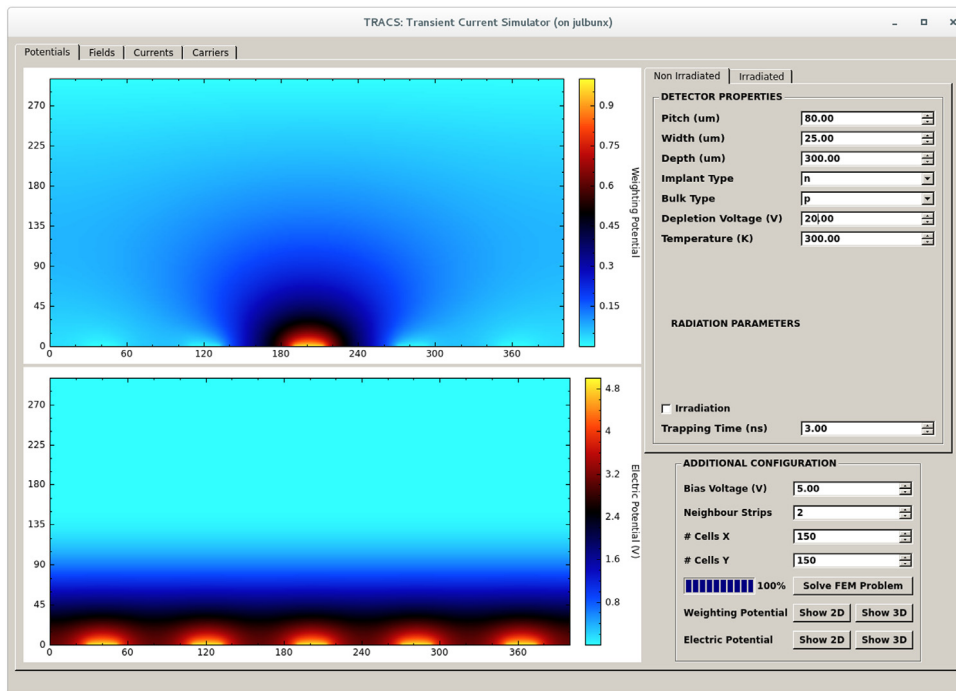


Fig. 2. Potentials label view of the TRACS GUI where weighting potential and drift potential graphical representations for a non-irradiated and non-depleted detector can be observed. Detector properties are defined on the right side.

Boltzmann constant and T is the temperature. The movement of each carrier follows a random walk that varies by collisions with the ionized impurity atoms and phonons of the network. In the absence of an external electric field, the average displacement from thermal motion is zero. However, if the carrier concentration in the semiconductor is not homogeneous, a net load movement occurs from the region of higher concentration to the region of lower concentration of carriers. In TRACS, diffusion is simulated following Einstein relations [22]:

$$D_{e,h} = \mu_{e,h} \frac{k_B T}{q} \quad (3)$$

where D is the diffusion constant that depends on temperature and μ the mobility of electrons or holes, respectively. After a time Δt , a displacement σ is calculated for each charge carrier using the diffusion constant with:

$$\sigma = \sqrt{2D\Delta t} \quad (4)$$

Diffusion phenomena can be switched on/off from the simulation by changing just one parameter in the configuration file.

4.3. Drift

If an electric field is present, charge carriers are accelerated among the mentioned collisions, a net directed movement is imposed on the thermal vibrations and a constant average velocity arises [22]. This velocity v is computed as the product of the mobility and the electric field:

$$v = \mu E \quad (5)$$

The trajectory of the particle can be determined by solving the following differential equation on the charge carrier position:

$$\frac{d\vec{x}(t)}{dt} = \vec{v} = \mu \vec{E}(\vec{x}(t)) \quad (6)$$

where μ is the mobility of the carrier which is different for electrons and holes and $\vec{x}(t)$ is the position of a charge carrier at a given time.

TRACS uses parameterization of the drift velocity according to Jacoboni model [23]:

$$v_{dr} = \frac{\mu_0}{(1 + (\frac{\mu_0 E}{v_{sat}})^\beta)^{1/\beta}} \cdot E \quad (7)$$

with v_{sat} the saturation velocity, μ_0 the low field mobility, and the exponent β . All these parameters are temperature dependent.

4.4. Signal generation

Any type of ionizing radiation that passes through the detector, creates electron–hole pairs in proportion to the energy deposited in the material. Charge carriers created in the space charge zone are immediately separated and transported towards the electrodes, pushed by the electric field. Therefore, signal is given by the current induced on the read-out electrodes by the movements of the electron–hole pairs, see Fig. 3.

The total current induced by an ionizing particle is computed by summing the contributions from both kind of moving charges. The total charge is proportional to the energy deposited by the ionizing particle. The current I_i induced over an electrode i by the movement of a carrier with charge q can be calculated following Shockley–Ramo’s theorem [25,26]. This approach is used in TRACS:

$$I_i = -q\vec{v}(x) \cdot \vec{E}_w(x) \quad (8)$$

where $v(x)$ is the drift velocity and depends on the applied electric field $E(x)$, x is the charge position, and E_w is referred to as the weighting field for a selected electrode. The weighting field derives from the weighting potential. The latter can be calculated by solving the Poisson’s equation with the following boundary conditions, readout electrode is active at 1 V, keeping all the others grounded, 0 V:

$$\Delta\phi_w = \nabla \cdot \vec{E}_w = 0 \quad (9)$$

The weighting field is different from the electric field and it only depends on the detector geometry.

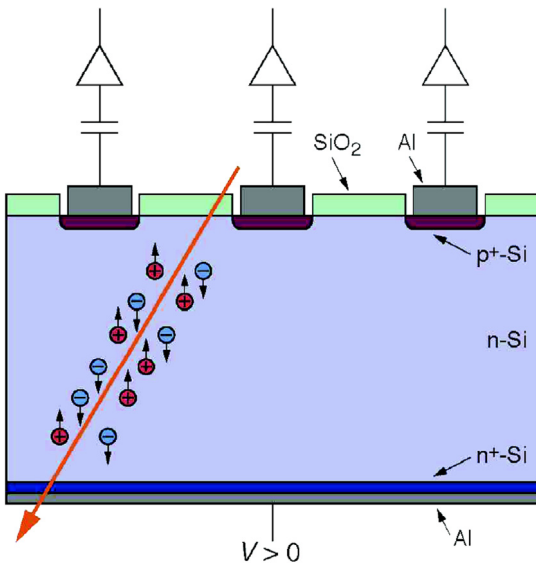


Fig. 3. Cross sectional view of a micro-strip detector with electron–hole pairs creation. Image reproduced from [24].

5. Numerical methods

To solve the Poisson’s equation, TRACS makes use of an open source software called FEniCS [27], implementing the finite elements method for discretization of the space. FEniCS expresses the Partial Differential Equation (PDE) as a variational problem, uses the mesh, the type of finite elements and the Boundary Conditions as inputs to solve for the electrostatic potential. Although the end user does not need to handle any details about the solving of these equations, a brief description of the process is given in the next paragraphs. Exhaustive documentation for FEniCS is found elsewhere [28].

The variational formulation of the problem is written in FEniCS high level Unified Form Language, which is nearly a transcription of the mathematical equations, in this case, of the Poisson’s equation. This description is general for the equation and it is written only once. The file is then compiled using the FEniCS form compiler which generates C++ code containing the implementation of the problem. Finally, an extra C++ file (main), coded by the TRACS developers, provides the details concerning the mesh, detector boundary conditions and space charge, that in turn the user provides via a steering file. The solving of Poisson’s equation is transparent to the end user.

5.1. Weighting potential and field

In order to calculate the weighting potential, Poisson’s equation is solved with zero space charge and new boundary conditions for the potential: 1 V in the collection electrode and 0 for all the remaining ones. The rest of parameters needed for the calculation are extracted from the configuration file at the beginning of the execution where the detector geometry is specified. Boundary conditions are defined taking into account the depletion region of the detector and they are recalculated if the bias voltage changes over the simulation (therefore changing the depleted region), as it is the case during simulated voltage scans. This is a very useful aspect when characterizing non-depleted detectors in order to investigate charge collection and electric field evolution as a function of voltage, until the detector reaches full depletion.

An example of weighting potential visualization for a given detector can be seen in Fig. 4.

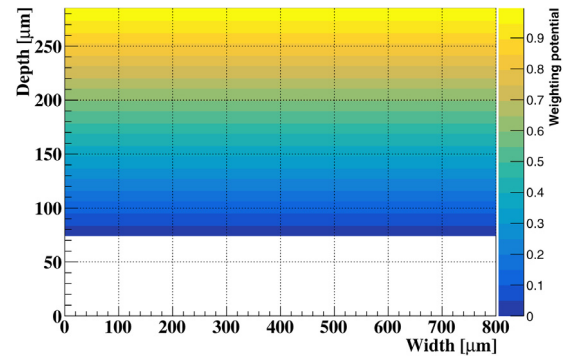


Fig. 4. Two dimensional TRACS simulation of the weighting potential of a diode at 20 V ($V_{dep} = 37$ V). The physical thickness of the detector is 285 μm, the simulated width being 800 μm. Due to under-depletion, the physical border of the detector for the calculation of the weighting potential appears to be inside the bulk.

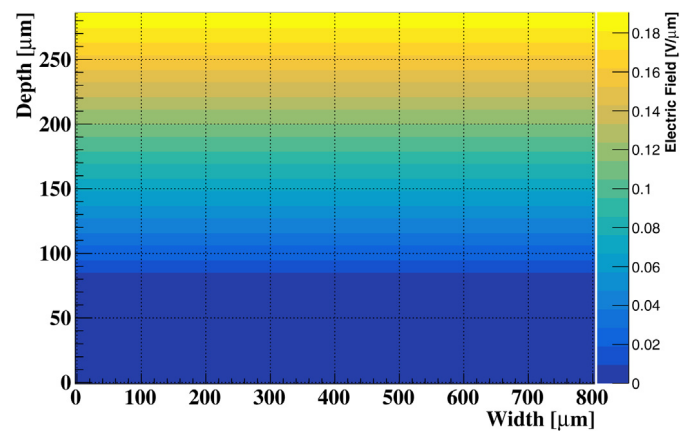


Fig. 5. TRACS simulation 2D of the electric field of a diode at 20 V, it fully depletes at 37 V. The physical thickness of the detector is 285 μm, the simulated width being 800 μm.

5.2. Electric potential and field

For non-irradiated detectors, a uniform space charge distribution produces an electric field that varies linearly in depth, reaching a maximum value at the p–n junction [22]. The effective doping concentration N_{eff} can be experimentally determined from capacitance measurements or estimated from measurements of collected charge as a function of voltage in TCT measurements. A picture of a calculated electric field for a simulated, non-irradiated detector, can be seen in Fig. 5.

For irradiated detectors, the leakage current induces a gradient of the effective doping concentration, leading to non-linear electric fields [29]. TRACS incorporates up to 3 different space charge parameterizations that are able to reproduce this behavior. Built-in N_{eff} models are: linear, tri-constant or trilinear. In the first one, the effective doping concentration varies linearly as a function of the depletion depth. In the second the space is divided into 3 regions and N_{eff} is piecewise constant. In the third, N_{eff} varies linearly within each region. To model, for instance, a tri-constant space charge, 5 parameters are needed. Namely: values of N_{eff} in each region, beginning and the end of the central region (the junction is placed at $z = 0$ and the detector depleted width is known from the applied voltage). When TRACS is used to fit simulation to real data, these N_{eff} parameters can be extracted from measurements (as well as the trapping time, see next subsection). Details on the usage of TRACS as a minimization tool and results on real detectors will be covered in a future communication.

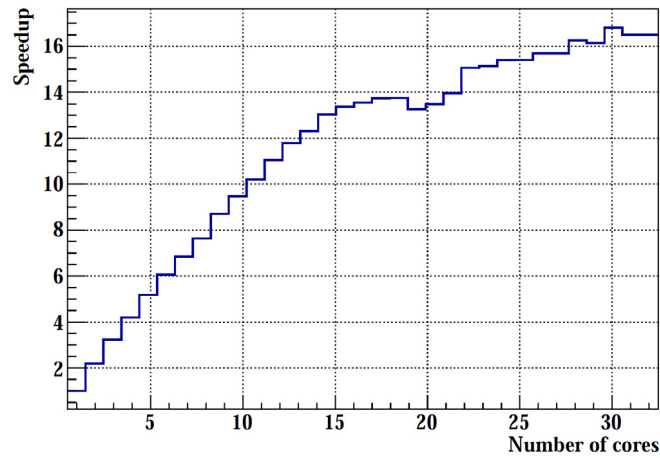


Fig. 6. Time reduced using parallelization: from 148 min (1 core) to 12 min (16 cores) over a simulation with 320 waveforms.

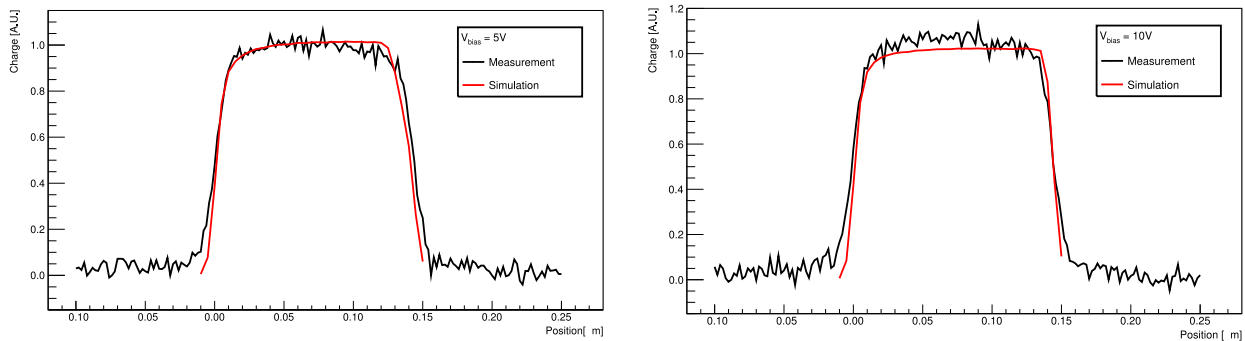


Fig. 7. Edge-TCT measured (black line) and simulated (red line) charge collection profiles, room temperature, 5 V and 10 V bias voltages for a non-irradiated detector n-on-p 150 μm thick diode. The diode is fully depleted at 7 V.

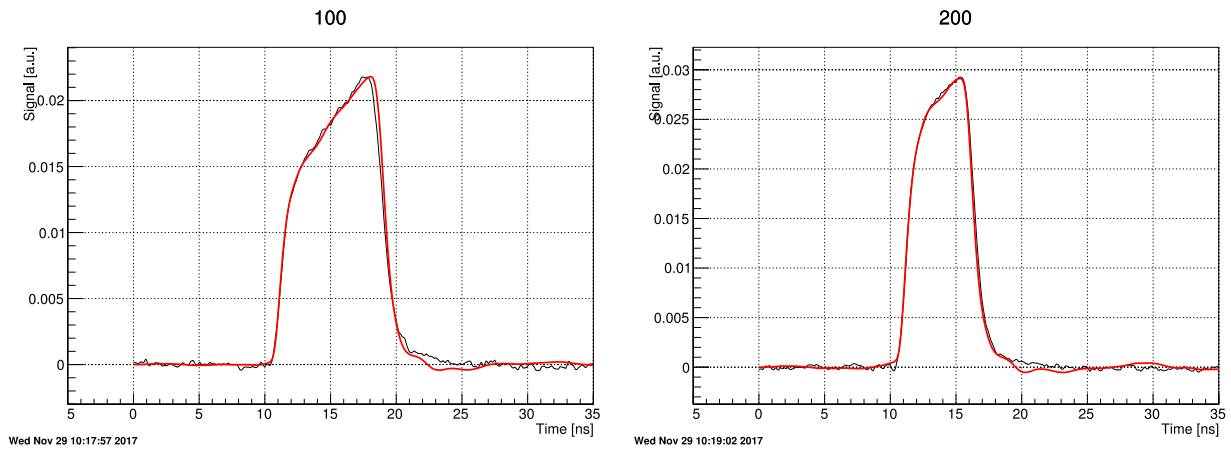


Fig. 8. Current transient measured (black line) and simulated (red line), room temperature, at 100 V and 200 V respectively for a non-irradiated detector with red laser light, bottom illumination.

5.3. Induced current

Once the electric field and the weighting field have been calculated in the depletion region of the detector, the final step of the simulation is to calculate the current induced by the movement of carriers. In the most general case, these excess carriers are simulated via an externally supplied script and saved to a text file. The file contains information on the type of carrier, and the generation position. The script allows to simulate typical illumination conditions (top/bottom, red/infrared) and customize laser parameters (penetration depth, Gaussian width, ...).

As explained in Section 4.4, TRACS uses Shockley–Ramo’s theorem for the induced current calculation. The current contribution of each carrier after a lapse Δt is calculated as long as the carrier is inside the detector boundaries or until the maximum simulation time is reached. We have decided to follow each carrier for the sake of flexibility. It is precisely in this part where we carry out the software parallelization, dividing the number of carriers among the number of available cores. On the other hand, we use adaptive integrators with arbitrary precision, Odeint C++ library, for the calculation of the renewed position of each carrier. Summarizing the complete process of induced current calculation:

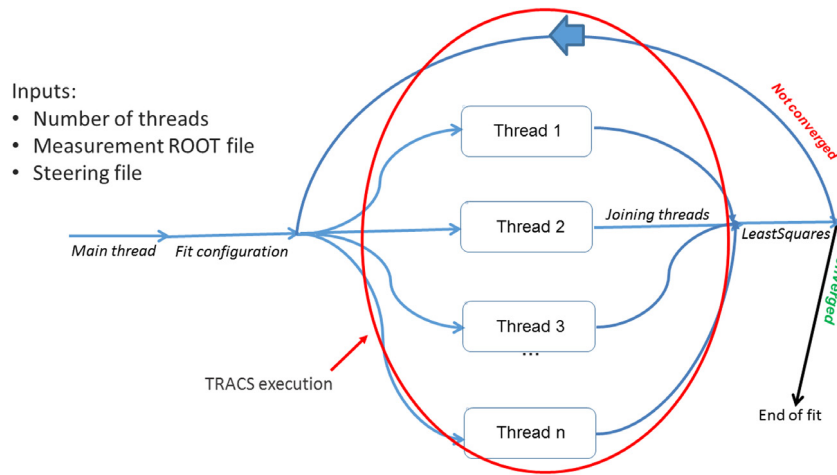


Fig. 9. Scheme generally describing the fitting process where TRACS is launched with a given number of threads after the wrapping of the user function for Minuit. Least squares calculation is carried out after each new TRACS execution.

- For each charge carrier, the program checks if it is inside detector boundaries. If it is out, the current contribution corresponding to the current time step Δt is equal to 0, but the carrier is still followed up, in case it re-enters the depleted region via diffusion. The tracking of carriers finishes once the maximum simulation time is reached.
- At each position, the 2D components of the electric and weighting field are looked up. The modulus of the electric field is used to calculate the mobility and drift velocity of the carrier. The value of the instantaneous current is updated adding the Ramo-current contribution at this position in time.
- The position of the carrier is renewed. In order to calculate the carrier movement due to drift, TRACS uses ODEINT [30], a C++ library for numerically solving ordinary differential equations, implementing a fourth-order *Runge–Kutta* iterative method.
- If diffusion phenomena is activated, the position of the carrier is again updated due to a random walk according to Eq. (4) smeared with a Gaussian function.

Finally, after following these stages for every carrier in the carrier list, the total current is computed summing all contributions from both types of carriers: $I_{i,total} = I_{i,e} + I_{i,h}$.

In irradiated detectors, defects created can act as trapping centers affecting the charge collection efficiency. Traps are mostly unoccupied in the depletion region due to the lack of free charge carriers and can then hold the signal carriers for a time longer than the charge collection time, this leads to a detriment of the charge collection efficiency [31]. In TRACS, trapping effects are parameterized as an exponential decay, being $\tau_{e,h}$ the effective trapping time for electrons and holes:

$$I_{trapping}(t) = I_{total}(t) \cdot e^{-\frac{t}{\tau_{e,h}}} \quad (10)$$

6. Multi-threading

Parallel computing is the concurrent use of multiple computing resources in order to solve a computational problem. The key part in any multi-thread program consists in finding the discrete parts of the execution that can be solved concurrently. Since TRACS can be used to fit real data, and minimizations demand hundreds or thousands of executions, parallelization was needed to obtain fitting results in affordable time. The parallel execution of the code works as well during normal execution of the code (non-fitting).

TRACS carries out the calculation of the induced current following the selected scan type chosen by the user: *edge*, *bottom* or *top*. For the calculation TRACS follows a given amount of carriers read from

a file specified by the user. The induced current calculation for each of these carriers is the discrete part of the execution that can be solved concurrently. The carrier distribution among threads is done following Round-robin scheduling to assure an exact workload for each thread. When TRACS starts its execution, it reads the number of cores set by the user or gets it automatically from the system.

Simulation of an edge-TCT measurement (15 ns total time, 50 ps time step, 37 000 simulated carriers), see Fig. 6, in a CERN virtual machine (32 cores at 2.4 GHz, Ubuntu Linux 14.04) shows a linear improvement of the execution time (expressed as speedup factor with respect to a single core) until 16 cores, then a slower gain up to 32 cores. The reason for the change in slope is due to the hardware topology of the system (to which we do not have physical access). Still, a remarkable sixteen times improvement in execution time is obtained with the maximum number of cores available in the virtual machine. A detailed analysis should be carried out to understand the behavior of the multi-threading execution beyond 16 cores in order to fine tune the scheduler policy or possible code dependencies.

Following these results, we advise the use of TRACS over other available tools when the user wants to perform complex simulations using two or more types of scan, or simply carry out fits. It is not possible to establish a direct comparison between TRACS and other tools except for a basic simulation case presented in Section 7 and summarized in Fig. 8. In this situation most of the tools behave the same, taking into account that the simulation takes only several seconds to run and remarkable differences cannot be measured. But it is, as explained above, in more complex simulations or fits where TRACS takes advantage, firstly because it exclusively allows such configurations explained in previous sections and secondly because it will always get a better performance from the machine where it is executed thanks to its multi-threading software.

7. Comparison of measurements with simulations

In order to validate TRACS, we have compared the simulation with laboratory measurements. TCT data was taken using the TCT+ system of the Solid State Detector (SSD) lab of the EP-DT group at CERN. Prior to these measurements, the device was characterized electrically, to obtain the load capacitance. This input is needed by the software, to simulate realistic waveforms and to estimate the depletion voltage of the detector.

Measured and simulated charge collection as a function of sensor depth for two different bias voltages of an edge-TCT scan of a diode are shown in Fig. 7. The detector was moved with respect to the beam in steps of 2 μm . When the IR laser illuminates outside of the depleted volume the charge collected is zero while inside the depleted volume the charge collected is maximum.

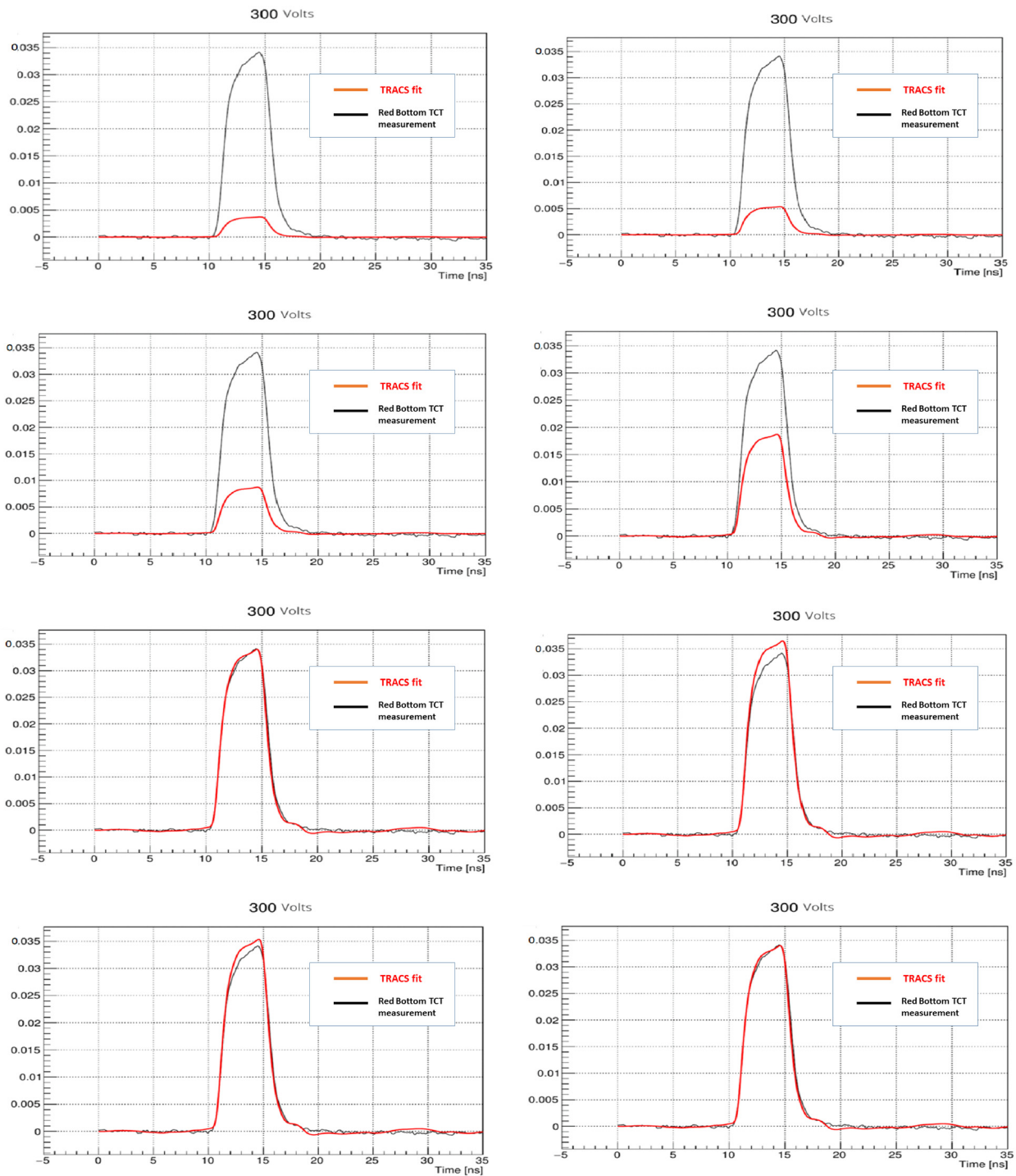


Fig. 10. Eight current transient representations of intermediate TRACS results of a fit, including the first estimation (top left) and the last one (bottom right). In red we can observe TRACS result getting closer to the given data representation, in black, a red bottom TCT characterization, 300 V bias voltage.

Fig. 8 compares the current transient for a bottom red measurement of a non-irradiated n-on-p 300 μm thick diode at 100 V and 200 V with the simulated transients after RC shaping.

8. Preliminary fit with TRACS

Once measurements for non-irradiated detectors are well described in simulations, we can proceed with fits. As mentioned before, the challenge is to fit the simulation to data and extract detector parameters measurements. This task is not affordable in TCAD within a manageable

time scale and we consider that this achievement contributes to differentiate TRACS from others fast simulators. Thus, the main objective of this point is just to demonstrate the correct functioning of the whole fit process that involves the coordinate execution of ROOT, Minuit and TRACS multithreading.

TRACS is now used as a library and will be called several times (tens to hundreds) during the fitting process. It is, indeed, within this functionality where the parallelization and the modular design of the code arise as the improvements that allow to efficiently achieve a fit.

A brief summary of the fitting process, detailed in Fig. 9, consist in launching TRACS including a file of measurements with induced current

waveforms in ROOT format, a steering file describing the same internal parameters of the measured detector and optionally, the number of cores to use. One internal parameter might be left as free parameter in order to be extracted, i.e, for non-irradiated detectors depletion voltage, capacitance or thickness.

The first step of the fitting process is to set up for the minimization. We create the minimization object in order to obtain an initial approximation of the χ^2 . Then, the function is wrapped for Minuit [32] and MIGRAD is used as a minimizer algorithm. Other possible choices such as simplex have been studied but they offer lower precision and local minimum.

The next step is to run TRACS as in a normal simulation, with the parameters defined in the steering file and an approximate value for the parameter that we want to extract. This execution is performed in parallel as explained in 6. When the execution of TRACS is finished, MIGRAD evaluates the result. If MIGRAD converges normally to a point, the program ends, otherwise TRACS execution is re-launched with a new estimation of the free parameter. It is Minuit itself who calculates the new value of the parameter and considers the moment in which the adjustment process must end, in case there is no convergence between the measurement and the simulation.

One of the most challenging part of this work is to understand and properly use Minuit and MIGRAD. We consider the fitting functionality as preliminary due to the difficulty of using these tools with complex functions and several parameters to fit. For the moment, we are leaving one unique internal parameter to be extracted while we tune the whole algorithm in order to manage the extraction of more parameters.

The next example shows the result of a fitting over a well known non-irradiated pin diode measured using red Bottom scan with the following parameters:

- Bias voltage = 300 V
- Temperature = 293 K
- W = 285 mm
- Fluence = 0 neq/cm²
- Capacitance = 4.17 pF
- Depletion voltage = 38 V
- Bulk = p

In the example shown, the result is known a priori and we know that Minuit should converge. We describe a simple case: the free parameter to fit is a value that relates the number of carriers of the data and the simulation. This value does not have a real impact when it comes to characterizing a detector, however it is extremely useful to check the whole machinery of the fitting. Parameters limits, error estimation and other mathematical aspects related to Minuit execution are not explaining since we consider they are out of scope of the goal of this section.

Fig. 10 shows several *snapshots* of the adjustment process that help to understand the functioning of the whole process and to observe its correct execution. There are 8 representations of the intermediate ROOT files generated by TRACS. They are obtained at the end of each execution and compared to the initial measurement file in ROOT. As shown, Minuit is getting a better fit every time, and once it finds a minimum it follows the same line search, it tests with nearby points to finally go back to the best solution found so far, which turns out to be a very good fit.

9. Conclusions

TRACS (TRAnsient Current Simulator) is a parallel software that calculates induced Ramo currents for any arbitrary distribution of charge carriers. It is able to simulate both diode and microstrip geometries of non-irradiated detectors.

TRACS shows an excellent agreement with laboratory measurements, using either depleted or non-depleted non-irradiated detectors. This validates the implementation of the $e-h$ transport model. TRACS executes concurrently over multi-cores machines reducing the

time of execution up to a factor 16 in 32 core machines. It also can be used as a library due to its modular design. These characteristics, multithreading and modularity, allow the software to properly interact with Minuit and ROOT to successfully carry out simple fits for diodes.

The program works either using a Graphical User Interface (GUI, normally used for visualization and quick feedback) or as a set of libraries (C++ interface) that can be used from an external user software. TRACS also comes with interface to Minuit [32] minimization package to fit simulated data to experiments.

As part of the ongoing work in detectors simulation, irradiation is introduced ad-hoc, as a spatial parameterization of the effective doping concentration. Three different models are built-in: 3 regions of (different) constant doping concentration, 3 regions of (different) doping concentration (linear variation within each region) and a linear change of N_{eff} across the bulk. A double junction like electric field shape can be retrieved from these profiles.

In the near future, it is expected to increase the complexity of the fits and to extract more than one parameter at a time.

TRACS is licensed under LGPL 3.0, this license allows developers and companies to use and integrate the software in their own programs.

Acknowledgments

This work was performed in the framework of the CERN-RD50 collaboration. This project has been partially supported by the Spanish Ministry of Economy and Competitiveness (MINECO) under grant number FPA2013-48387-C6-1-P and FPA2015-71292-C2-1-P. We thank the CERN summer student program for financial support of 3 students participating in this project.

References

- [1] L. Rossi, O. Brüning, High Luminosity Large Hadron Collider: A description for the European Strategy Preparatory Group, Technical Report, CERN-ATS-2012-236, CERN, Geneva, 2012.
- [2] ATLAS, The ATLAS Experiment at the CERN Large Hadron Collider, J. Instrum. 3 (2008) S08003.
- [3] CMS, The CMS experiment at the CERN LHC, J. Instrum. 3 (2008) S08004.
- [4] B. Schmidt, The high-luminosity upgrade of the LHC: Physics and technology challenges for the accelerator and the experiments, J. Phys. Conf. Ser. 706 (2016) 022–002.
- [5] RD50, <http://www.cern.ch/rd50>, 2002.
- [6] G. Lindström, et al., Radiation hard silicon detectors - Developments by the RD48 (ROSE) collaboration, Nucl. Instrum. Methods Phys. Res. A 466 (2001) 308–326.
- [7] G. Lindström, et al., Developments for radiation hard silicon detectors by defect engineering - Results by the CERN RD48 (ROSE) Collaboration, Nucl. Instrum. Methods Phys. Res. A 465 (2001) 60–69, Cited By 73.
- [8] M. Moll, Development of radiation hard sensors for very high luminosity colliders, cern-rd50 project, in: Proceedings of the 11th International Workshop on Vertex Detectors, Nucl. Instrum. Methods Phys. Res. A 511 (2003) 97–105.
- [9] J. Calvo, M. Fernández, P. de Castro, A. Diez, U. Senica, M. Moll, I. Vila, 2014, <https://github.com/CERN-SSD/TRACS>.
- [10] Synopsys, 2018, <https://www.synopsys.com/silicon/tcad.html>.
- [11] Silvaco, 2018, <https://www.silvaco.com/>.
- [12] Cogenda, 2018, <https://www.cogenda.com/>.
- [13] G. Kramberger, 2017, <http://kdetsim.org/>.
- [14] CERN, 2017, <https://root.cern.ch/>.
- [15] F. Cenna, et al., Weightfield2: A fast simulator for silicon and diamond solid state detector, in: Proceedings of the 10th International Conference on Radiation Effects on Semiconductor Materials Detectors and Devices, Nucl. Instrum. Methods Phys. Res. A 796 (2015) 149–153.
- [16] G. Panjkovic, et al., Modeling and simulation of the entire detector system by using Matlab and Simulink, in: IEEE Nuclear Science Symposium Medical Imaging Conference, pp. 1168–1172.
- [17] A. Ellakany, et al., Modeling and simulation of a hybrid 3D silicon detector system using SILVACO and Simulink/MATLAB framework, in: 2016 28th International Conference on Microelectronics (ICM), pp. 377–380.
- [18] D. Pohl, 3D-silicon and passive CMOS pixel-detectors for the ATLAS pixel detector upgrade, Slides from RD50 Workshop Krakow, 2017.
- [19] Z.L.V. Eremin, W. Chen, Analytical Solutions of Minimum Ionization Particle induced Current Shapes of Silicon Detectors and Simulation of Charge Collection Properties, in: IEEE Nuclear Science Symposium Medical Imaging Conference.
- [20] G. Kramberger, et al., Investigation of irradiated silicon detectors by Edge-TCT, IEEE Trans. Nucl. Sci. 57 (2010) 2294–2302.

- [21] G. Kramberger, et al., Electric field modeling in heavily irradiated silicon detectors based on Edge-TCT measurements, PoS Vertex 2012 (2013) 022.
- [22] H. Spieler, Semiconductor Detector Systems, in: Series on Semiconductor Science and Technology, OUP Oxford, 2005.
- [23] C. Jacoboni, C. Canali, G. Ottaviani, A.A. Quaranta, A review of some charge transport properties of silicon, Solid-State Electron. 20 (1977) 77–89.
- [24] M. Krammer, et al., Silicon detectors, Slides from CERN EDIT School, 2010.
- [25] W. Shockley, Currents to conductors induced by a moving point charge, J. Appl. Phys. 9 (1938) 635–636.
- [26] S. Ramo, Currents induced by electron motion, Proc. IRE 27 (1939) 584–585.
- [27] M.S. Alnæs, et al., The FEniCS Project Version 1.5, Arch. Numer. Softw. 3 (2015).
- [28] G.N.W.e.a. Ander Logg, Kent-Andr Mardal, Automated Solution of Differential Equations By the Finite Element Method, Springer, 2012.
- [29] E. Verbistkaya, V. Eremin, Z. Li, The origin of double peak electric field distribution in heavily irradiated silicon detectors, Nucl. Instrum. Methods Phys. Res. A 476 (2002) 556–564.
- [30] K. Ahnert, M. Mulansky, 2012, <http://headmyshoulder.github.io/odeint-v2/>.
- [31] M. Moll, Radiation damage in silicon particle detectors: Microscopic defects and macroscopic properties (Ph.D. thesis), Hamburg U., 1999.
- [32] F. James, Minuit, in: Proceedings of the 1972 CERN Computing and Data Processing School 72-21, 1972.